Soft Computing and Machine Intelligence

Soft Computing and Machine Intelligence Journal Vol (X), Issue (X), 20XX

PiperNet: The novel algorithm that outperforms accuracy

Talha, Mehedi Hasan [a], Chowdhury, Saqib Jahir [b], Shawon,Bm Rahful Hasan [c], and Siddique, Al Amin [d]

(a) 19-39597-1 (a) Section: B
(b) 18-38643-3 (b) Section: B
(c) 18-38270-2 (c) Section: B
(d) 18-38270-2 (d) Section: B
* Corresponding author: Talha, Mehedi Hasan - iamtalha.mht@gmail.com

Abstract: Deep Learning is a machine learning method that uses non-linear transformation to find features in large amounts of data. It is now widely used for supervised learning in a variety of fields. Since 2012, Convolutional Neural Network (CNN) has been the best technique for image classification. Keras is a popular API for neural networks written in Python that can also be used in R for users considering deep learning models for real-world applications. We attempt to investigate Deep Neural Network parameter estimation procedures and CNN model structures ranging from basic to advanced techniques. We also use Keras to try to figure out some critical steps in CNN that can improve image classification performance in the Skin Cancer ISIC dataset. On our findings, our newly built model performed better than VGG16 model on this dataset. There are four convolutional layers and a flatten layer in this model.

Keywords: CNN; Model; Deep Learning; ISIC; Keras;

1. Introduction

With the growing amount of data and computing resources, many machine learning algorithms have evolved as a major component of artificial intelligence (AI). Statistical models and algorithms use patterns and conclusions to achieve specified tasks. Deep neural networks (DNN), extreme gradient boosting (XGBoost), and support vector machines (SVMs) are decision tree-based approaches that are significant for real regression and classification challenges. We'll concentrate on deep neural networks, often known as "deep learning," in this report. Deep learning is a sophisticated technique to machine learning that incorporates human brain knowledge, statistics, and applied mathematics. It isn't a new method that has been proposed. Deep learning is made up of numerous layers of artificial neural networks (ANNs) that connect each node to artificial neurons similar to those found in the biological brain. You can find high-level characteristics in the input data by solving difficult non-linear equations and stacking numerous hidden layers between the input and output layers. Several deep neural columns become experts on inputs preprocessed in different ways; their predictions are averaged. [1]

2. Related Works

The supervised representation learning issue has received a lot of attention in the area of photographs and other types of imagery. K-means is frequently used in unsupervised learning, such as data clustering. After the data has been clustered, it may be utilized to improve classification accuracy. When working with photos, you may hierarchically group image patches to get a better representation of those images. Another typical method is to utilize an auto-encoder stack, which then decodes the code to recreate the picture. In this method, feature representations learnt from image pixels have also been proven. Deep neural networks have also been shown to enhance a hierarchical model. Numerous research on generative picture models have been undertaken, with two categories commonly distinguished: parametric and non parametric. Texture generation, super-resolution, and in-painting are all done with non parametric models. Many scientists have investigated parametric models that create images. But on the Skin Cancer ISIC dataset, there has been a few works done. On the Kaggle site, the recent upload had only a test accuracy of 20%.

3. Proposed Model

The model we created performs well. This is referred to as PiperNet. Severeal convolutional layers were used. Everything that was used, helped getting the accuracy higher. Further information will be found in section 4.2.

4. Result

4.1. Data Representing

Our model has one of the best accuracy on the Skin Cancer ISIC dataset. We simply worked with presenting the data correctly at first. As it can be seen on 1, the images were represented with no errors.

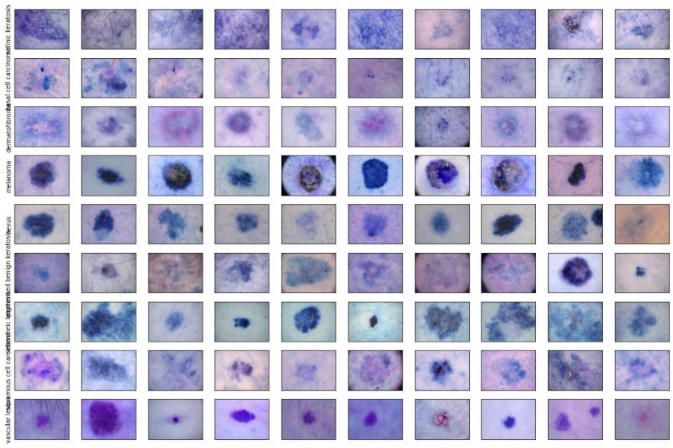

Figure 1. **Data Representation**

4.2. Building the Model

After representing the images, we reshaped the images into 32X32 size. This reshaping was necessary otherwise we would not be able to train the data accurately as the images were differently shaped. Once reshaping done, we went on to build the model. On our model, there are four convolutional layers. We used Tanh activation on three of the convloutional layers and ReLu on the last one. It helped to give us a training accuracy over 92%. Nadam optimizer was used in this model. And the model was sequential. Another thing that gave the boost to the training accuracy was batch normalization. Batch Normalization allows us to use much higher learning rates and be less careful about initialization. [2]

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_4 (Conv2D)            (None, 32, 32, 32)        896
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 16, 16, 32)        0
_________________________________________________________________
batch_normalization_4 (Batch (None, 16, 16, 32)        128
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 16, 16, 64)        18496
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 (None, 8, 8, 64)          0
_________________________________________________________________
batch_normalization_5 (Batch (None, 8, 8, 64)          256
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 8, 8, 64)          36928
_________________________________________________________________
max_pooling2d_6 (MaxPooling2 (None, 4, 4, 64)          0
_________________________________________________________________
batch_normalization_6 (Batch (None, 4, 4, 64)          256
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 4, 4, 64)          36928
_________________________________________________________________
max_pooling2d_7 (MaxPooling2 (None, 2, 2, 64)          0
_________________________________________________________________
batch_normalization_7 (Batch (None, 2, 2, 64)          256
_________________________________________________________________
flatten_1 (Flatten)          (None, 256)               0
_________________________________________________________________
dense_1 (Dense)              (None, 9)                 2313
=================================================================
Total params: 96,457
Trainable params: 96,009
Non-trainable params: 448
_________________________________________________________________
```

Figure 2. Model

4.3. Training

Training is the most important part of the model. On this model, training was conducted on a very convenient way. The main concern was to get the highest possible accuracy.

```
Epoch 1/5
56/56 [==============================] - 1s 12ms/step - loss: 0.2542 - accuracy: 0.9280 - val_loss: 1.9095 - val_accuracy: 0.38
39
Epoch 2/5
56/56 [==============================] - 1s 12ms/step - loss: 0.2388 - accuracy: 0.9341 - val_loss: 1.8718 - val_accuracy: 0.39
73
Epoch 3/5
56/56 [==============================] - 1s 12ms/step - loss: 0.2279 - accuracy: 0.9330 - val_loss: 1.9087 - val_accuracy: 0.40
18
Epoch 4/5
56/56 [==============================] - 1s 12ms/step - loss: 0.2206 - accuracy: 0.9347 - val_loss: 1.8793 - val_accuracy: 0.38
84
Epoch 5/5
56/56 [==============================] - 1s 12ms/step - loss: 0.2133 - accuracy: 0.9330 - val_loss: 1.9150 - val_accuracy: 0.40
18
```

Figure 3. Training

As it can be seen from 3, the accuracy of the epochs conducted was over 90%. It is one of the highest accuracy on this particular dataset in Kaggle. This model easily beats the VGG16 CNN model.

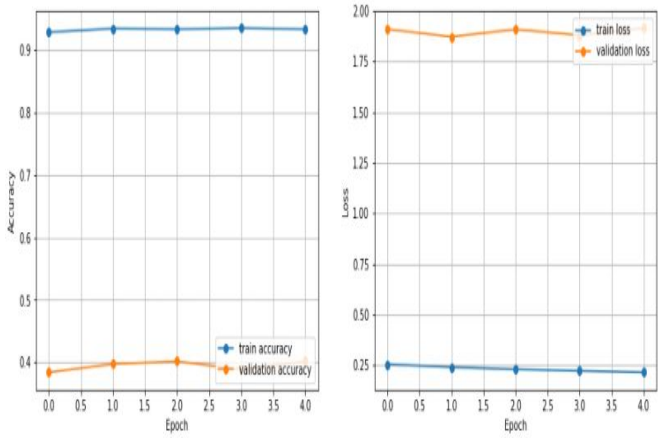

Figure 4. Accuracy Graph

4.4. Testing

Testing is the last phase. Our testing result is also satisfying. The testing accuracy was over 32%. Which is one of the best on Skin Cancer ISIC dataset.

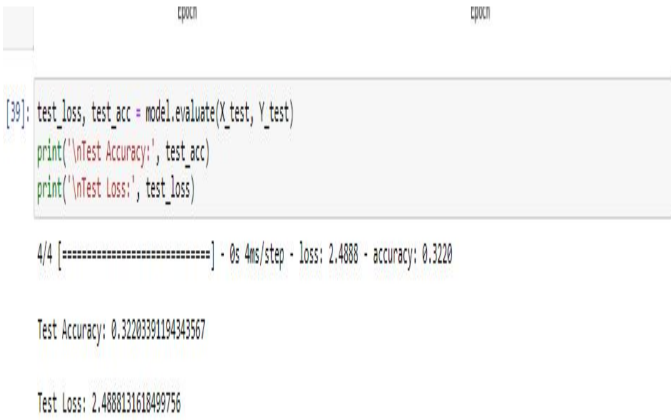

Figure 5. **Testing Accuracy**

4.5. Contribution

Table 1. Contribution Table.

Name	**Contribution**
Talha, Mehedi Hasan	Worked with the model, visualizing and report
Chowdhury, Saqib Jahir	Worked with data representing
Shawon,Bm Rahful Hasan	Worked with researching and helped coding
Siddique, Al Amin	Worked with project report and helped coding

5. Discussion

Deep Convolutional Neural Networks have proved exceptional performance in object recognition, detection, and localization, as well as a variety of other computer vision tasks. Despite all of the advances demonstrated by the various proposed architectures, there were few insights and logical reasoning about how they had achieved state-of-the-art records, making further improvements subject to trial and error strategies. Our model is surely one of the best models, as the accuracy rate is higher than the VGG16 model. But it needs improvements. The accuracy could be much higher if the algorithm was more efficient. Over the time, it will be more efficient.

6. Conclusions

Deep learning has gained popularity as computer resources and data sizes have increased. Keras uses core functions to help those people quickly become accustomed to deep learning methods. Deep neural networks are useful for supervised learning with unusual data such as images and texts. We organized deep learning concepts for statisticians in terms of parameter estimation procedures and advanced techniques that improve model performance. To better understand the procedure, we describe the computation of a deep network in matrix form. We also concentrated on the convolutional neural network, which is the best at dealing with image data, and discovered that it outperforms other machine learning methods that cannot learn local features.

References

1. ACires¸an D, Meier U, and Schmidhuber J (2012). Multi-column deep neural networks for image classification, arXiv:1202.2745.
2. Ioffe S and Szegedy C (2015). Batch normalization: Accelerating deep network training by reduc-ing internal covariate shift. In Proceedings of the 32nd International Conference on MachineLearning,37, 448–456, arXiv:1502.03167.